RECEIVED
JUN 26 2001
Technology Center 2100

RECEIVED
JUN 26 2001
Technology Center 2100

Bill Wilhelm
06/20/2001 09:40 AM

To: shelley beckstrand
cc:
From: Bill Wilhelm/Endicott/IBM@IBMUS
Subject:

Excerpt from Microsoft Visual Studio Version 6.0 Help function on "EnterCriticalSection" function for mutual exclusion processing.

EnterCriticalSection
The EnterCriticalSection function waits for ownership of the specified critical section object. The function returns when the calling thread is granted ownership.

```
VOID EnterCriticalSection(
  LPCRITICAL_SECTION lpCriticalSection   // pointer to critical
                                         // section object
);
```

Parameters
lpCriticalSection
Pointer to the critical section object.
Return Values
This function does not return a value.

Remarks
The threads of a single process can use a critical section object for mutual-exclusion synchronization. The process is responsible for allocating the memory used by a critical section object, which it can do by declaring a variable of type CRITICAL_SECTION. Before using a critical section, some thread of the process must call the InitializeCriticalSection or InitializeCriticalSectionAndSpinCount function to initialize the object.

To enable mutually exclusive access to a shared resource, each thread calls the EnterCriticalSection or TryEnterCriticalSection function to request ownership of the critical section before executing any section of code that accesses the protected resource. The difference is that TryEnterCriticalSection returns immediately, regardless of whether it obtained ownership of the critical section, while EnterCriticalSection blocks until the thread can take ownership of the critical section. When it has finished executing the protected code, the thread uses the LeaveCriticalSection function to relinquish ownership, enabling another thread to become owner and access the protected resource. The thread must call LeaveCriticalSection once for each time that it entered the critical section. The thread enters the critical section each time EnterCriticalSection and TryEnterCriticalSection succeed.

Once a thread has ownership of a critical section, it can make additional calls to EnterCriticalSection or TryEnterCriticalSection without blocking its execution. This prevents a thread from deadlocking itself while waiting for a critical section that it already owns.

Any thread of the process can use the DeleteCriticalSection function to release the system resources that were allocated when the critical section object was initialized. After this function has been called, the critical section object can no longer be used for synchronization.

If a thread terminates while it has ownership of a critical section, the state of the critical section is undefined.

QuickInfo
Windows NT: Requires version 3.1 or later.
Windows: Requires Windows 95 or later.
Windows CE: Requires version 1.0 or later.
Header: Declared in winbase.h.
Import Library: Use kernel32.lib.

See Also
Synchronization Overview, Synchronization Functions, DeleteCriticalSection, InitializeCriticalSection, InitializeCriticalSectionAndSpinCount, LeaveCriticalSection TryEnterCriticalSection

G.W. Wilhelm, Jr.
Electronic Media Management System - Subsystem Development
(607)755-8943 t/l 855-8943
Internet: wilhelm@us.ibm.com